Robust Real-time Vision-based Aircraft Tracking From Unmanned Aerial Vehicles

Changhong Fu , Adrian Carrio , Miguel A. Olivares-Mendez ,
Ramon Suarez-Fernandez , Pascual Campoy

I. INTRODUCTION

Visual aircraft tracking has been researched and developed fruitfully in the robot community recently. However, real-time robust visual tracking for arbitrary aircraft (also referred to visual *aircraft model-free* tracking), especially in Unmanned Aerial Vehicle (UAV), remains a challenging task due to significant appearance change, variant surrounding illumination, partial aircraft occlusion, blur motion, rapid pose variation, and onboard mechanical vibration, low computation capacity and delayed information communication between UAVs and Ground Station (GS).

In the literatures, many visual trackers have obtained the promising tracking performances for arbitrary aircrafts, where, the morphological filtering technology as the most popular method has been applied in many vision-based Sense-and-Avoid (i.e. See-and-Avoid) systems, e.g. A. Wainwright et al [1], T. Gandhi et al [2], R. Carnie et al [3] and J. S. Lai et al [4]. However, a big number of false positives will be generated by this approach, and it requires the reliable morphological operators to adaptively detect the aircraft under different backgrounds. Although D. Dey et al [5] utilize shape descriptor and SVM-based classifier to reduce false positives, however, it should be trained offline with hand-labeled samples in large amounts of image data.

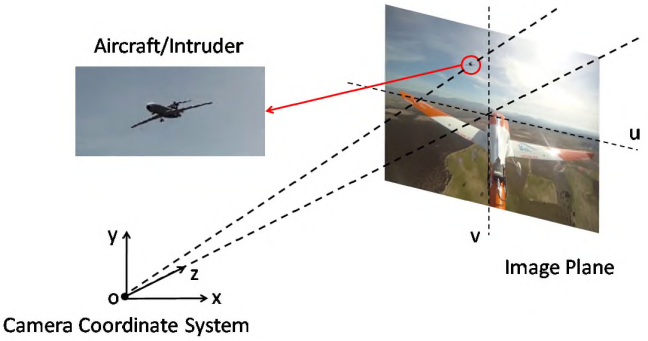


Fig. 1: Vision-based aircraft inspection on UAV, where, the monocular camera sensor is fixed on the tail of UAV.

J. W. McCandless [6] presented an optical flow method for aircraft detection, and A. Mian [7] proposes a modified KLT tracking algorithm to track aircrafts, which uses a feature clustering criterion to track aircraft based on its multiple local features, and this local features are continuously updated to make the tracker robust to appearance changing of the aircraft. However, all these methods can be generally categorized as the generative-based method, and they did not use the valuable background information to improve the tracking performances [8].

In this paper, we apply the discriminative-based algorithm (also called visual *tracking-by-detection* method) to track aircraft/intruder in the midair using UAVs, which employ an adaptive binary classifier to separate the aircraft from background during frame-to-frame tracking, and online Multiple-Instance Learning (MIL) method [9] has been used to handle the ambiguity problem, which put the positive samples and negative ones into positive and negative bags, respectively, and then trains a classifier in an online manner using bag likelihood function. This method has demonstrated good performance to handle drift, and can even solve significant appearance changes in the cluttered background.

Moreover, we adopt Multi-Resolution (MR) strategy to cope with the problems of strong motions (e.g. onboard mechanical vibration) or large displacements over time. Additionally, this strategy can help to deal with the problems that are the onboard low computational capacity and information communication delays between UAVs and Ground Station

(GS). Using this strategy, especially in the Multi-Classifier voting mechanism, the importances of test samples have been used to reject samples, i.e. the lower resolution features are initially applied in rejecting the majority of samples (called Rejected Samples (RS)) at relatively low cost, leaving a relatively small number of samples to be processed in higher resolutions, thereby ensuring the real-time performance and higher accuracy.

To the author's best knowledge, this visual tracker has not been presented for solving the online learning and tracking arbitrary aircraft problems in the UAVs. The proposed AM^3 tracker runs at real-time frame rates and also performs favorably in the midair collision warning and avoidance evalutaion system for UAVs in terms of efficiency, accuracy and robustness.

II. DISCRIMINATIVE VISUAL TRACKING

A. Preliminaries

Discriminative Visual Tracking (DVT) takes the tracking problem as a binary classification task to separate target from its surrounding background. A generic process of the DVT is presented in the Algorithm 1.

Algorithm 1 DVT.

Input: the kth frame

1. Extract a set of image samples:
$\mathcal{S}^{\alpha}$=$\{\mathbf{S}|\|\mathbf{l}(\mathbf{S})-\mathbf{l}_{k-1}\|<\alpha\}$, where, $\mathbf{l}_{k-1}$ is the target location at (k-1)th frame, and online select feature vectors.
2. Use classifier trained in the (k-1)th frame to these feature vectors and find the target location $\mathbf{l}_k$ with the maximum classifier score.
3. Extract two sets of image samples:
$\mathcal{S}^{\beta}$=$\{\mathbf{S}|\|\mathbf{l}(\mathbf{S})-\mathbf{l}_k\|<\beta\}$ and $\mathcal{S}^{\gamma,\delta}$=$\{\mathbf{S}|\gamma<\|\mathbf{l}(\mathbf{S})-\mathbf{l}_k\|<\delta\}$, where, $\beta<\gamma<\delta$.
4. Online select the feature using these two sets of samples, and update the classifier.

Output: (1) Target location $\mathbf{l}_k$
(2) Classifier trained in the k-th frame

In the Algorithm 1, the parameter α is called search radius, which is used to extract the test samples in the kth frame, the parameter β is the radius applied for extracting the positive samples, while the parameter γ and δ are the inner and outer radii, which are used to extract the negative samples.

However, the ambiguity problem can confuse the classifier. P. Viola et al [10] used a Multiple-Instance Learning (MIL) [11] approach to solve this ambiguity problem in face detection task successfully.

B. Tracking with Online Multiple-Instance Learning

Recently, B. Babenko et al [9] also presented an online Multiple-Instance Learning (MIL) algorithm, i.e. MIL tracker, to track the targets robustly. In this paper, we adopted this method for visual aircraft tracking, as shown in Figure 2. And the Algorithm 2 shows the pseudo code of this tracker.

In the Algorithm 2, the posterior probability of sample $\mathbf{S}_{ij}$ to be positive, i.e. $p(y=1|\mathbf{S}_{ij})$, is computed by the

Algorithm 2 MIL.

Input: Dataset $\{\mathcal{S}_i,y_i\}_{i=0}^{1}$, where $\mathcal{S}_i=\{\mathbf{S}_{i1},\mathbf{S}_{i2},...\}$ is the ith bag, and $y_i\in\{0,1\}$ is a binary label of sample $\mathbf{S}_{ij}$

1. Update weak classifier pool Φ=$\{h_1,h_2,...,h_M\}$ with data $\{\mathbf{S}_{ij},y_i\}$
2. Initialize $H_{ij}=0$ for all i,j
3. **for** k=1 to K **do**
4. Set $\mathcal{L}_m$ =0, m=1, ..., M
5. **for** m=1 to M **do**
6. **for** i=0 to 1 **do**
7. **for** j=0 to N+L-1 **do**
8. p_{ij}^m=$\sigma(H_{ij}+h_m(\mathbf{S}_{ij}))$
9. **end for**
10. p_i^m=1-$\prod_j(1-p_{ij}^m)$
11. $\mathcal{L}_m\leftarrow\mathcal{L}_m+y_i log(p_i^m)+(1-y_i)log(1-p_i^m)$
12. **end for**
13. **end for**
14. m^*=$argmax_m(\mathcal{L}_m)$
15. $h_k(\mathbf{S}_{ij})\leftarrow h_{m^*}(\mathbf{S}_{ij})$
16. $H_{ij}=H_{ij}+h_k(\mathbf{S}_{ij})$
17. **end for**

Output: Classifer $H_K(\mathbf{S}_{ij})=\sum_k h_k(\mathbf{S}_{ij})$, and $p(y=1|\mathbf{S}_{ij})$=$\sigma(H_K(\mathbf{S}_{ij}))$

Bayesian theorem, $\sigma(z)=1/(1+e^{-z})$ is a sigmoid function, the strong classifier H_K is constructed by selected K weak classifiers, i.e. H_K=$\sum_{k=1}^{K}h_k$. And $\mathcal{L}$ is the bag log-likehood function: $\mathcal{L}=\sum_i(y_i log p_i+(1-y_i)log(1-p_i))$.

For each image sample, it is represented as a vector of Haar-like features [12], which is denoted by function $\mathbf{f}(\mathbf{S}_{ij})=(f_1(\mathbf{S}_{ij}),f_2(\mathbf{S}_{ij}),...,f_K(\mathbf{S}_{ij}))^T$. Each feature consists of 2 to 4 rectangles, and each rectangle has a real valued weight. The feature value is then a weighted sum of the pixels in all the rectangles.

We assume that Haar-like features in $\mathbf{f}(\mathbf{S}_{ij})$ are independently distributed and assume uniform prior $p(y=0)=p(y=1)$. Then, the classifier $H_K(\mathbf{S}_{ij})$ is described with the Haar-like feature $\mathbf{f}(\mathbf{S}_{ij})$ as

$$H_K(\mathbf{S}_{ij})=ln\left(\frac{p(\mathbf{f}(\mathbf{S}_{ij})|y=1)p(y=1)}{p(\mathbf{f}(\mathbf{S}_{ij})|y=0)p(y=0)}\right)=\sum_{k=1}^{K}h_k(\mathbf{S}_{ij}) \tag{1}$$

where,

$$h_k(\mathbf{S}_{ij})=ln\left(\frac{p(f_k(\mathbf{S}_{ij})|y=1)}{p(f_k(\mathbf{S}_{ij})|y=0)}\right) \tag{2}$$

and

$$\begin{aligned}p(f_k(\mathbf{S}_{ij})|y_i=1)&\sim N(\mu_1,\sigma_1),\\ p(f_k(\mathbf{S}_{ij})|y_i=0)&\sim N(\mu_0,\sigma_0)\end{aligned} \tag{3}$$

The update schemes for the parameters μ_1 and σ_1 are:

$$\begin{aligned}\mu_1&\leftarrow\eta\mu_1+(1-\eta)\frac{1}{N}\sum_{j|y_i=1}f_k(\mathbf{S}_{ij})\\ \sigma_1&\leftarrow\eta\sigma_1+(1-\eta)\sqrt{\frac{1}{N}\sum_{j|y_i=1}(f_k(\mathbf{S}_{ij})-\mu_1)^2}\end{aligned} \tag{4}$$

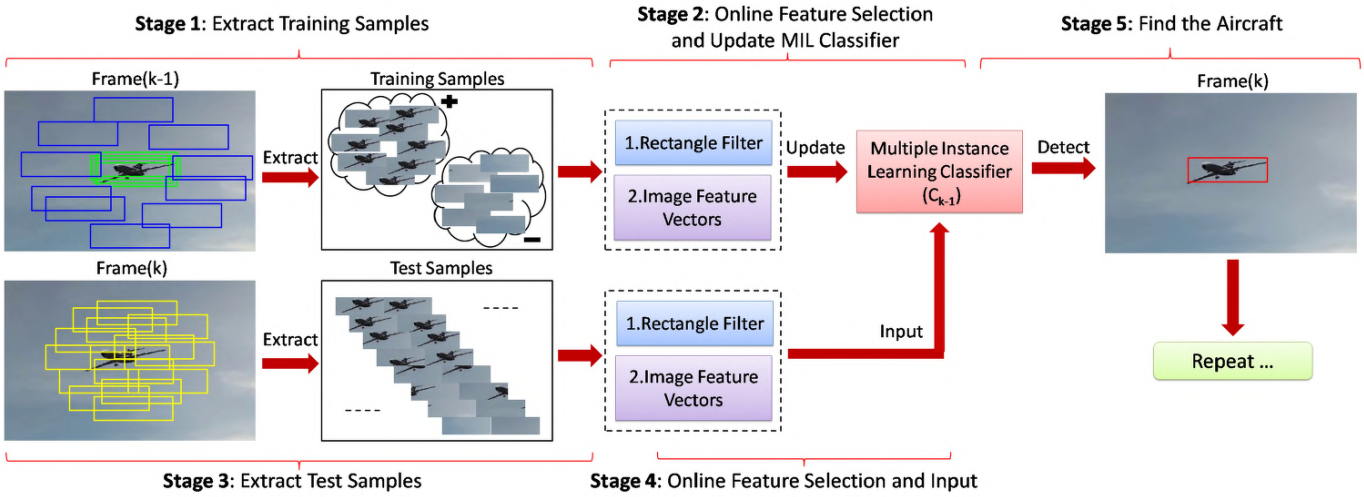


Fig. 2: Visual aircraft tracking via Multiple-Instance Learning (MIL). The adaptive MIL classifier is updated with online boosting features in the (k-1)th frame, and then applied to estimate the aircraft location in the kth frame.

where, N is the number of positive samples and η is a learning rate parameter. The update schemes for μ_0 and σ_0 have similar formulas.

TABLE I: Relationship between Search Radius (α) and Number of Extracted Test Samples (N_S)

Radius α	Sample N_S	Radius α	Sample N_S
30	2809	17	889
29	2617	16	793
28	2449	15	697
27	2285	14	609
26	2109	13	517
25	1941	12	437
24	1789	11	373
23	1649	10	305
22	1513	9	249
21	1369	8	193
20	1245	7	145
19	1125	6	109
18	1005	5	69

III. HIERARCHY-BASED TRACKING STRATEGY

A. Hierarchy-based Tracking

Although the discriminative-based approaches often achieve superior tracking results, and tolerate the motions in the range of search radius, but for the tracking on-board UAV, we have observed that discriminative visual tracking algorithms are sensitive to the strong motions or large displacements. The search radius for extracting test samples can be set to be larger, as shown in Algorithm 1, to get more tolerance for these problems, however, more test samples (including noises) will be generated, which influence the real-time and accuracy performances, as shown in TABLE I. Therefore, Multiple Resolution (MR) approach was proposed to deal with these problems, as shown in Figure 3, which also can help to deal with the problems that are the onboard low computational capacity and information communication delays between UAVs and Ground Station (GS).

B. Configurations

1) Number of Pyramid Levels (N_{PL}): Considering the images are downsampled by a ratio factor 2, the Pyramid Levels of the MR structure are defined as a function below:

$$N_{PL} = \lfloor log_2 \frac{min\{\mathbf{T}_W, \mathbf{T}_H\}}{minSizes} \rfloor \quad (5)$$

where, $\lfloor * \rfloor$ is the largest integer not greater than value $*$, $\mathbf{T}_W$, $\mathbf{T}_H$ represent the width and height of target $\mathbf{T}$ in the highest resolution image (i.e. the lowest-level of pyramid: 0 level), respectively. And $minSizes$ is the minimum size of target in the lowest resolution image (i.e. the highest-level of pyramid: p_{max} level, $p_{max} = N_{PL}$-1), in order to have enough information to estimate the motion model in that level. Thus, if the $minSizes$ is set in advanced, the N_{PL} directly depends on the width/height of tracking target $\mathbf{T}$.

2) Motion Model (l) Propagation: Taking into account that the motion model estimated in each level is used as the initial estimation of motion for the next higher resolution image, therefore, the motion model propagation is defined as follows:

$$\mathbf{l}_k^{p-1} = 2\mathbf{l}_k^p \quad (6)$$

where, p represents the pth level of the pyramid, $p = \{p_{max}, p_{max} - 1, ..., 0\} = \{N_{PL} - 1, N_{PL} - 2, ..., 0\}$, and k is the kth frame.

3) Number of Rejected Sample (N_R) : Since the MR approach provides the computational advantage to analyze features and update classifiers in low resolution images, the majority of samples will be rejected based on their classifier scores (i.e. sample importances) in the lower resolution image, leaving a fewer number of samples to be processed in the higher resolution image. Thus, the tracker obtains higher tracking speed, better accuracy than a single full (high) resolution-based adaptive tracker, the rejected sample number is defined as:

$$N_R^p = \xi^p N_S^p \quad (7)$$

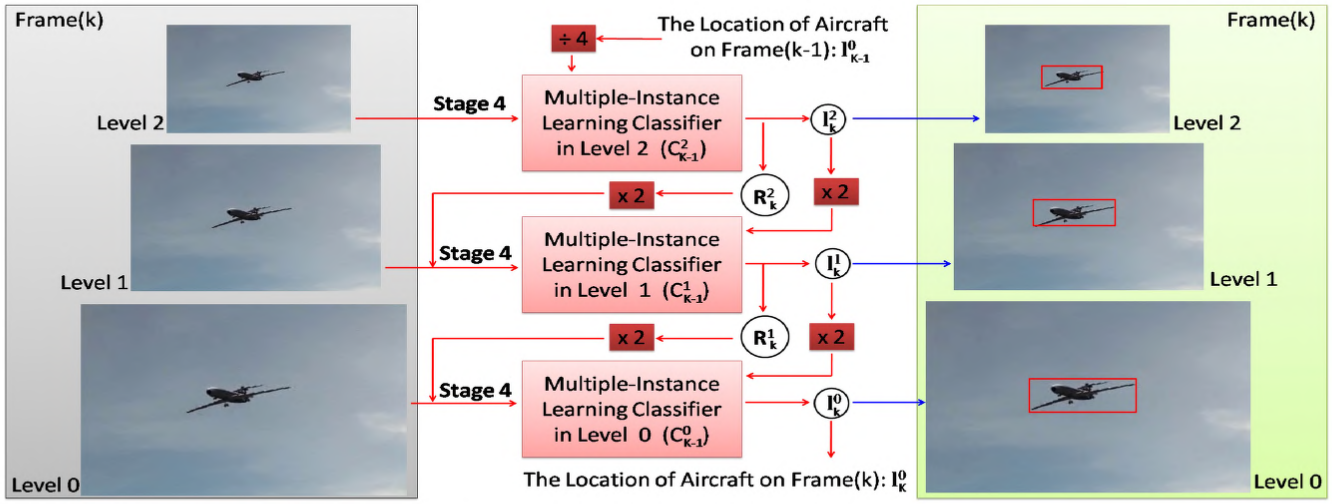


Fig. 3: AM^3 visual tracker. The kth frame is downsampled to create MR structure. The lower resolution features are initially used to reject the majority of samples at relatively low cost, leaving a relatively small number of samples to be processed in higher resolutions. The C^p_{k-1} represents the adaptive classifier updated in the pth level of pyramid of (k-1)th frame.

where, p represents the pth level in the pyramid, ξ^p is the reject ratio ($0 < \xi^p < 1$), and N^p_S is the number of test samples. Especially, the sample with maximum score in the rejected samples is the Boundry Sample (B^p_k).

4) Search Radius Propagation: The euclidean distance between the location of B^p_k and $\mathbf{l}^p_k$ is the Recursive Distance (R^p_k), which will be propagated to next higher resolution image as the search radius:

$$\alpha^{p-1}_k = 2R^p_k \tag{8}$$

where, p represents the pth level in the pyramid, and k is the kth frame.

Figure 4 and TABLE I show the details of our presented tracker, which are the confidence maps constructed by importances of test samples from non-hierarchical (a) and hierarchical (b,c,d) tracking results in the kth frame. We assume that the tracker requires radius 20 (in pixels) to search the aircraft in the full (high) resolution frame, then 1245 samples will be extracted to test with classifier, however, with our tracker just need a small number of samples (371 in total) within different resolution frames, and obtains higher accuracy, as shown in Figure 5.

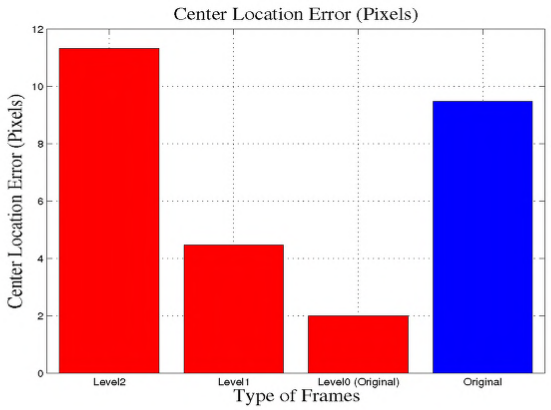


Fig. 5: Comparision of Center Location Errors in the kth frame. Red and Blue Bars represent the hierarchical and non-hierarchical tracking results, respectively.

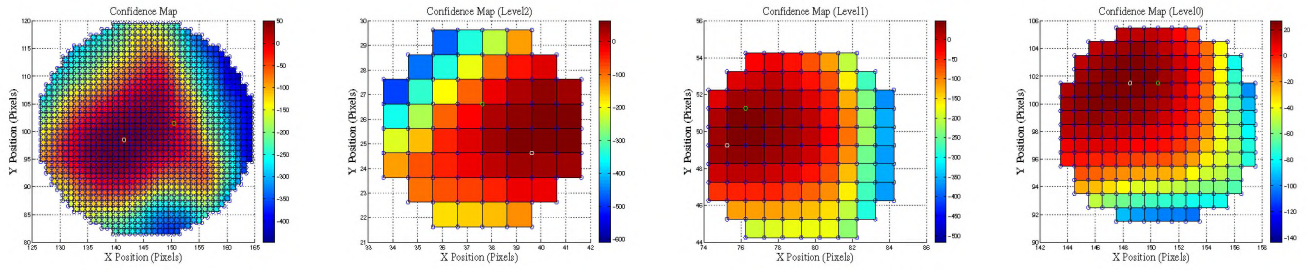
IV. VISION-BASED AIRCRAFT TRACKING

A. Midair Collision Visual Warning and Avoidance Evaluation System

Vision-based aircraft detection and avoidance algorithms demand real scenario images to be tested. These images sometimes are difficult or dangerous to obtain, especially for detecting collision course. For this reason, a new midair collision visual warning and avoidance evaluation system has been developed, this system allows the user to define any flight trajectories and backgrounds using different aircrafts/intruders, where real world images took from some UAVs are fused with virtual images containing 3D aircraft model. These virtual images are obtained taking into account scene illumination, camera vibrations and lens distortions, thereby producing the very realistic video stream.

The 3D pose and attitude of aircraft are pre-defined frame-by-frame, therefore, the performances of different tracking algorithms can be evaluated and compared. The main part of system in software is accomplished with three steps. Firstly, image vibration information is collected from the real world images. Secondly, the virtual image of an aircraft/intruder 3D model is constructed. Finally, both real frames and vitrual images are fused.

1) Real Image Vibration Information Collection: Due to the existence of vibrations in the real world images, this image vibration effects should be reproducted in the virtual images in order to obtain the most realistic results. The virtual image is transformed according to the homography transformation, which is a (3×3) matrix that links coordi-



(a) α = 20, N = 1245 (b) α = 5, N = 69 (c) α = 6, N = 109 (d) α = 8, N = 193

Fig. 4: Confidence Maps. They are constructed by the importances of test samples (Blue Circle) in the kth frame, where, the Green Circle represents the Ground Truth (GT), the White Circle is the tracking result. And (a) is the non-hierarchical tracking result, (b)(c)(d) are the hierarchical tracking results in different resolution frames, i.e. low, middle and high resolutions.

nates between two views of the same scene, i.e.:

$$x_i' = Hx_i \tag{9}$$

The homography matrices that map the relationship between the first and the other consecutive frames are obtained with below processes:

- Corner feature extraction from the first frame
- Optical flow calculation on the new frame
- Homography matrix collection using RANSAC

2) Virtual Image Construction: In order to obtain a virtual image displaying an aircraft, a 3D virtual scenario is generated using OpenGL. A virtual camera system and a virtual 3D aircraft are placed and orientated, where, the virtual camera system is configured with the same angle of view in the on-board real camera system, and the virtual 3D aircraft is constructed using a 3D geometry model of the aircraft and a texture, which allows the 3D model to have a realistic appearance.

Additionally, the 3D secene is rendered with a green background, which allows to easily distinguish the aircraft pixels from the background pixels, i.e. chroma key technique[1].

3) Real and Virtual Image Fusion: The original background image is undistorted and backwarped so that the subsequent warping and distortion applied to both the aircraft and the background will help to generate an unaltered background. Performing the fusion with this way, the interpolation during the warping and distortion processes will produce a more realistic result. The fusion results are shown in the Figure 1, 2, 3, 6a and 7a with a common commercial plane: Boeing 727.

B. Comparisions in Evaluation System

In this section, we compared our AM^3 tracker with 3 latest state-of-art trackers (Frag[2] [13], TLD[3] and MIL[4]) on two different types of challenging situations: (I) *Cloudy*; (II) *Strong light*. The performances of these trackers were evaluated with the Ground Truth (GT), as shown in the Figure 6b, 6c, 7b and 7c. And the Center Location Error (CLE) is used to be the evaluation measurement [14], which is defined as the Euclidean distance from the detected aircraft center to its ground truth center at each frame, as shown in the Figure 6d and 7d.

1) Test 1: Comparision under the cloudy background: This situation contains four main challenging factors: (I) Strong motions (e.g. onboard mechanical vibration and wind influence) or large displacements; (II) Scale change; (III) Illumination Variation; (IV) Background Clutters.

For the tracking performances, as shown in Figure 6b, 6c and 6d, Frag tracker lost its target firstly when the aircraft was flying from the non-cloud area to the cloud area. While the TLD tracker also lost its target when the illumination of aircraft is similar to the edge of cloud. MIL can track its aircraft well at the beginning, however, it also lost the aircraft when the target was flying from cloud area to non-cloud area. Our new proposed AM^3 can locate the aircraft in all evaluation processes, and the performances of these four trackers have been shown in the TABLE II.

2) Test 2: Comparision under the strong light background: This situation also contains three main challenging factors: (I) Strong motions (e.g. onboard mechanical vibration and wind influence) or large displacements; (II) Scale change; (III) Illumination Variation.

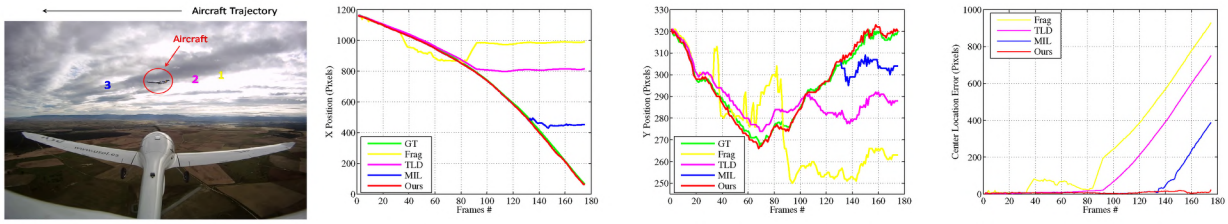
During the tracking process, as shown in Figure 7b, 7c and 7d, the Frag tracker lost its target when a small cloud confused it, as the yellow **1** shown in Figure 7a. For TLD tracker, it is able to relocate on the target at the beginning, but it lost the aircraft completely from the 85th frame. For the MIL tracker, it prones to locate the tail of aircraft, but it also lost the aircraft when the aircraft was flying from the non-strong light area to the strong light area. Our new presented visual tracker AM^3 can track the aircraft all the time until the aircraft flow out of the FOV.

The Center Location Error (CLE) (in pixels) for these two evaluations in this paper is shown in below Table:

TABLE II: Center Location Error (in pixels)

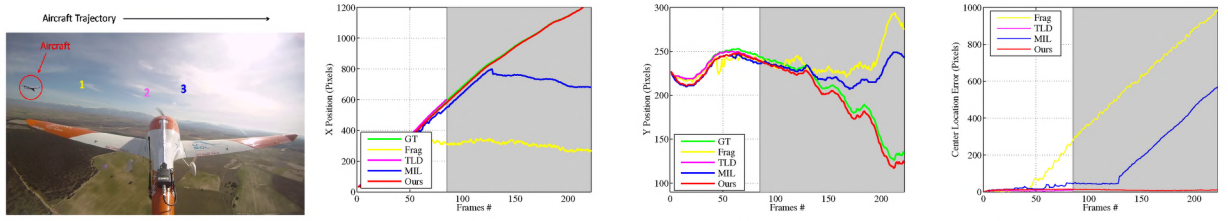
Situations-Trackers	Frag	TLD	MIL	AM^3
Cloudy	275	172	48	**7**
Strong Light	425	NaN	154	**10**

[1]http://en.wikipedia.org/wiki/Chroma_key
[2]http://www.cs.technion.ac.il/~amita/fragtrack/fragtrack.htm
[3]http://gnebehay.github.io/OpenTLD/
[4]http://vision.ucsd.edu/~bbabenko/project_miltrack.shtml



(a) Trajectory and Lost Locations. (b) X Position Errors. (c) Y Position Errors. (d) Center Location Errors.

Fig. 6: Visual aircraft/intruder tracking on-board UAV under *Cloudy* background (Frame Size: 1280×960), where, No.1 (Yellow) represents the lost location tracked by Frag tracker. For TLD and MIL trackers, their lost locations are marked with No.2 (Pink) and No.3 (Blue), respectively. Their tracking performances are evaluated with Ground Truth (Green).



(a) Trajectory and Lost Locations. (b) X Position Errors. (c) Y Position Errors. (d) Center Location Errors.

Fig. 7: Visual aircraft/intruder tracking on-board UAV under the *Strong Light* background (Frame Size: 1280×720), where, the Grey Shadows show that the TLD tracker lost the aircraft/intruder completely.

V. CONCLUSIONS AND FUTURE WORKS

This paper proposed a new real-time visual tracker named AM^3 to track an arbitrary aircraft/intruder in the midair, and test results in the evaluation system show that it outperforms the existing state-of-art trackers in different kind of challenging situations in terms of robustness, efficiency and accuracy.

In the future works, we will compare with more existing state-of-art trackers in the midair collision warning and avoidance evalutaion system using different backgrounds, aircrafts and trajectories, and IMU/GPS data also will be used to compare with these visual trackers. Finally, the multiple aircrafts/intruders tracking algorithm will be developed.

ACKNOWLEDGMENT

The work reported in this paper is the consecution of several research stages at the Computer Vision Group-Universidad Politécnica de Madrid. This work has been sponsored by the Spanish Science and Technology Ministry under the grant CICYT DPI2010-20751-C02-01, the E-Vision Project (TSI-020100-2011-363), the OMNIWORKS project (an experiment funded in the context of the ECHORD project (FP7-ICT-231143)) and the China Scholarship Council (CSC). The authors would like to thank project E-Vision's coordinator (USol Company) to provide aerial test images, as well as the reviewers for valuable feedback and input.